观察题目可以知道，要想让有匹配的，则k。

我们把这些数按从小到大的顺序放进去的话，

假设我们要把数i放到A集合中，要满足：

1. 当A集合的数比B集合的数少的话， B集合中必须有一个没有被匹配的数要小于等于i-k才能放。

2. 否则， i肯定可以放在A中（让B后来放更大的数就行了）。

而小于等于i-k的个数只要管它有几个就行了，因为它们一定都可以和i匹配。

设a表示A集合中有几个数，b表示B集合中有几个数，

c表示i-k+1到i-1这些数的状态（0表示在A集合，1表示在B集合），cnt[c]表示c有几位为1。

假如我们要把第a+b+1个数放入A集合，那么上面的信息条件可以转化为：

1. a<b且a<=b-cnt[c]-1。

2. a>=b。

解释一下第一个为啥是b-cnt[c]-a>=1：

cnt[c]其实就是a+b+1-k+1（最小的不能与a+b+1匹配的数）到a+b这些数有几个数在B中,那么b-cnt[c]就表示1到a+b+1-k这些数有几个在B集合中，再减去a就是算出与A集合的数比配后剩几个。必须要剩不少于一个才能匹配a+b+1

用代码写就是这样：

|  |
| --- |
| bool Check(int a,int b,int c) {  return (a==n)?0:a>=b||b-cnt[c]-a>=1;//A集合满了就不能放  } |

设f[i][j][k]表示放入了1至i，A集合中有j个数，状态k的方案数。

i+1放入A中：f[i][j][k]转移至f[i+1][j+1][((k<<1))&sum]

i+1放入B中：f[i][j][k]转移至f[i+1][j][(k<<1|1)&sum]

完整代码：

|  |
| --- |
| #include <bits/stdc++.h>  using namespace std;  const int mod=1000000007;  int ans,f[101][51][512]= {1},cnt[512],n,m,sum;  bool Check(int a,int b,int c) {  return (a==n)?0:a>=b||b-cnt[c]-a>=1;  }  void GetMod(int &a,int b) {  if((a+=b)>=mod)  a-=mod;  }  int main() {  scanf("%d%d",&n,&m);  m--;  sum=(1<<m)-1;  for(int i=1; i<=sum; i++)  cnt[i]=cnt[i^(i&(-i))]+1;//统计i的二进制数有多少个1  for(int i=0; i<=2\*n-1; i++)  for(int j=0; j<=n; j++)  for(int k=0; k<=sum; k++) { //枚举所有方案  if(Check(j,i-j,k))  GetMod(f[i+1][j+1][((k<<1))&sum],f[i][j][k]);  if(Check(i-j,j,sum-k))//放到B中的条件和A差不多  GetMod(f[i+1][j][(k<<1|1)&sum],f[i][j][k]);  }  for(int i=0; i<=sum; i++)  GetMod(ans,f[2\*n][n][i]);  printf("%d\n",ans);  return 0;  } |